Human Body Shape Completion with Implicit Shape and Flow Learning

Boyao Zhou Di Meng Jean-Sébastien Franco Edmond Boyer
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
{boyao.zhou, di.meng, jean-sebastien.franco, edmond.boyer}@inria.fr

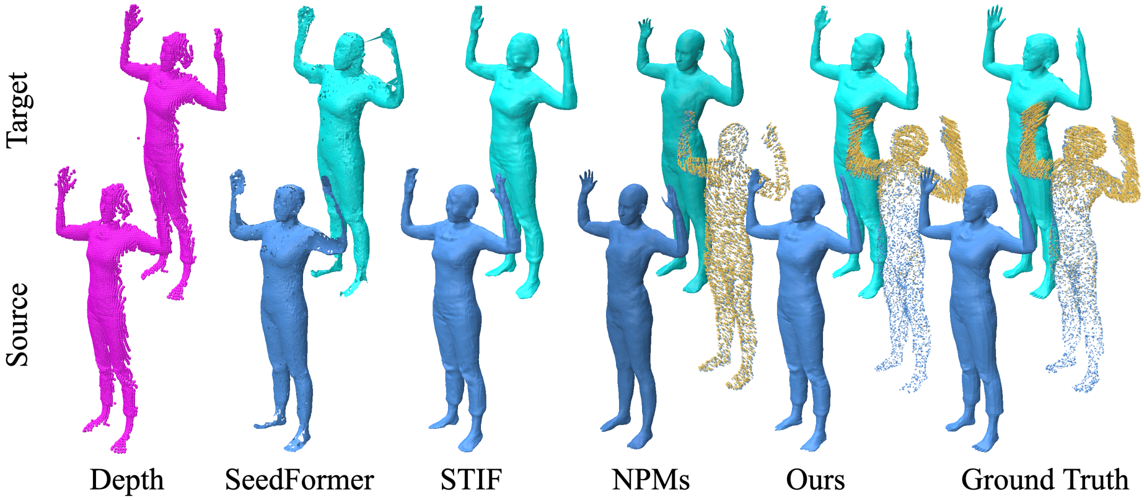

Figure 1. Human shape completions with various methods given 2 depth images. NPMs [31] and our approach provide also motion flows.

Abstract

In this paper, we investigate how to complete human body shape models by combining shape and flow estimation given two consecutive depth images. Shape completion is a challenging task in computer vision that is highly underconstrained when considering partial depth observations. Besides model based strategies that exploit strong priors, and consequently struggle to preserve fine geometric details, learning based approaches build on weaker assumptions and can benefit from efficient implicit representations. We adopt such a representation and explore how the motion flow between two consecutive frames can contribute to the shape completion task. In order to effectively exploit the flow information, our architecture combines both estimations and implements two features for robustness: First, an all-to-all attention module that encodes the correlation between points in the same frame and between corresponding points in different frames; Second, a coarse-dense to fine-sparse strategy that balances the representation ability and the computational cost. Our experiments demonstrate that the flow actually benefits human body model completion. They also show that our method outperforms the state-of-the-art approaches for shape completion on 2 *benchmarks, considering different human shapes, poses, and clothing.*

1. Introduction

The inference of human body shape information from depth observations has become a standard problem in computer vision. Depth sensors are now common and enable the digitization of humans using every day devices such as tablets or mobile phones, in turn opening a way to new consumer applications that build on this ability, *e.g.* virtual try on or avatar applications. Solving the problem efficiently is however difficult given human body observations that are, by construction, incomplete with a single frame. Considering several frames over time, as often available, can however improve the body shape estimation, provided that temporal consistency is effectively exploited. In this paper we consider how to build complete human shape models given these partial depth observations. Particularly we investigate how the combination of shape and motion flow estimations can benefit such shape completion tasks.

Different strategies for shape completion have been explored that exploit various priors over human shapes. Parametric body models such as SMPL [24] can be used as in [5, 31, 34]. The strong prior assumed with a parametric model ensures spatially and temporally coherent human shape predictions. However these predictions are inherently restricted to limited shape spaces. The preservation of the geometric details that can be present in depth maps, *e.g.*

face attributes or cloth wrinkles, is arduous. Other strategies build on weaker priors, relying on learning to characterize shapes and their completion. Early contributions in this respect [9, 40, 52] explore encoder-decoder network architectures with 3D convolutions and successfully predict complete distance fields in explicit voxel grids. They were subsequently extended to implicit representations that can provide continuous 3D shape functions such as occupancy [7,28], or distance fields [8,32], with limited memory costs compared to explicit voxel representations. Furthermore, temporal features provided by depth map sequences can also be accounted for with implicit representations that then become spatio-temporal [58]. Yet, without explicit correspondences over time, learning based methods can only partially exploit temporal consistency.

Such correspondences are encoded in the motion field between the input depth maps. This field, the scene flow, is traditionally estimated pixel-wise as an extension of the 2D optical flow. Recent learning-based strategies [22, 50] generally focus on observed points only and do not target shape estimation nor completion. Closer to this objective, OFlow [30] proposes a 4D model that combines shape and flow information in an implicit continuous representation. While the method can account for point clouds, it does not easily extend to shape completion with depth maps. Furthermore the shape and flow are estimated independently whereas we advocate a combined estimation of both.

To this aim, we propose a learning-based approach that considers two consecutive depth images as input and estimates a continuous complete representation of both shape occupancy (SDF) and motion as implicit functions, leveraging their representational advantages demonstrated for both problems independently. Our experiments show that such a combined estimation benefits the shape completion task with results that outperform existing works on standard datasets. The proposed approach is pyramidal and considers image features that are extracted in a coarse to fine manner, preserving both local and more global shape properties. In addition, with the aim to enforce consistency in both spatial and temporal domains, we take inspiration from the scene flow work [47] and introduce an all-to-all attention mechanism that accounts for spatial and temporal correlations between points in the two frames considered. Comprehensive ablation tests demonstrate the individual contributions of the pyramidal framework and attention mechanisms. Experiments were conducted on DFAUST [6] and CAPE [26] with both undressed and dressed humans. We provide comparisons with the state-of-the-art approaches for both shape and flow estimations and show consistent shape completion improvements with our method.

Method	Shape Completion	Continuous Shape Rep.	Continuous Flow Rep.	Detail Preservation	Scene Represent.
SceneFlow [22,47,50]	✗	✗	✗	✗	Points
4DComplete [19]	✓	✗	✗	✓	Voxels
STIF [58]	✓	✓	✗	✓	Implicit
NPMs [31]	✓	✓	✗	✓	Para. model
OFlow [30]	✓	✓	✓	✗	Implicit
Ours	✓	✓	✓	✓	Implicit

Table 1. Classification of related methods with respect to their abilities to: handle partial inputs; provide continuous shape and flow representations; preserve geometric details in the observations.

2. Related Works

We focus on works that solve for full shape completion given partial depth image observations. They roughly fall into two main categories: model-based and learning-based strategies. Let us also mention fusion-based methods [29, 53–55, 57] that typically update a canonical model with moving cameras but do not handle unobserved shape parts.

Model-Based: A robust strategy to obtain a complete shape model given partial observations is to rely on a low dimensional parametric shape space, typically built using principal component analysis, within which the observations are fitted. For instance, Weiss *et al.* [49], MoSh [23] and MoSh++ [27] recover full undressed body shapes from sparse inputs using the parametric models SCAPE [2] BlendSCAPE [14] or SMPL-H [36]. In order to handle more generic body shapes [1, 5, 33, 46], one can model cloth as an offset from the undressed parametric model SMPL [24]. This improves the modeling of clothed human shapes with some restrictions on the shape topology and on the ability to preserve high-frequency shape details imposed by the underlying shape space. Recently, neural parametric models, *e.g.* LBS-AE [17] and NPMs [31], have been proposed which encode full body representations into low dimensional latent spaces with auto-encoders. Partial data fitting can then be performed through the optimization of a data term with respect to the latent representation. While improving the expressivity of the shape space, these methods still build on global shape priors with hence limited abilities to preserve local shape details or to model arbitrary human shapes, as illustrated in our experiments.

Learning-based: Inspired by the success of 2D convolutional networks, early learning-based methods [9, 40, 52] consider 3D convolutions and predict distance fields in explicit voxel grids, given partial observations. In order to improve the precision-complexity tradeoff, neural implicit methods were proposed where decoders model continuous implicit functions that consider 3D locations and the associated encoder features as inputs to predict occupancies [7,28] or distances [3,12]. These methods achieve physically plausible results with a moderate cost during training but struggle with high-frequency details [28, 30] or rely on complete observations and full supervisions [3, 12, 48]. In this category, point-based methods were also introduced [21,44,51,56,59] that learn the mapping between complete and partial shape point clouds with the benefit of being able to handle more general human shape topologies with low memory costs, using point based representations. With the aim to benefit from observations over time when available, STIF [58] and H4D [15] use spatio-temporal implicit functions or motion priors for human motion modeling. While accounting for the time dimension, the proposed methods do not model explicit temporal correspondences, such as the flow, hence only partially exploit the time dimension. Our experiments demonstrate the benefit of introducing the flow in the estimation. A related method in that respect is the Occupancy-Flow network [30] which considers temporal sequences and infers a canonical shape from the first frame which is then deformed over the sequences with flows that are predicted in a 3D grid. However, the proposed cascaded structure does not enable flow features to benefit the shape estimation.

3. Method

To address the shape completion problem with a high level of detail, a popular and time-tested strategy is to rely on coarse-to-fine schemes, which have been applied to many domains including that of depth map completion [58], monocular shape estimation [39], or 2D optical flow [42]. A key aspect of our work is to simultaneously examine the shape completion and scene flow, where pyramidal cost-volume approaches have been very successful. Yet to apply these in our context quickly leads to a dimensionality problem, because cost-volume approaches combinatorially examine every motion possibility from consecutive feature volumes, which quickly leads to a bottleneck for high resolution volumes necessary to high detail recovery.

For this reason, in our approach we propose two distinctive coarse and fine paths to estimate shape and flow: on one hand a coarse path for which classical 3D convolutions and cost volumes are tractable to efficiently evaluate global shape and flow features (§3.2), and on the other hand, a fine path where we decisively leverage sparse 3D convolutions and flow cost combinations to infer high resolution details without paying the dimensionality penalty (§3.3).

In turn, the contained nature of the representation allows us to explore other combinatorially limitative improvement schemes such as attention techniques, which allow to enhance the model with global knowledge such as symmetries in the shape and flow inference (§3.4). We couple these aspects with a fully implicit SDF shape and flow representation, which has been many times shown successful to further the goal of detailed and continuous inference (§3.5). One of the key advantages of coupling our SDF extraction with our hybrid dense/sparse hierarchical scheme is that it alleviates inference difficulties in multi-modal, self-occlusion depth completion situations which hinder previous approaches, *e.g.* [38, 58] that only decode the SDF using a single projected depth feature. To achieve the required multimodality in depth, both of our coarse and fine inference paths rely on a first stage of pyramidal depth feature encodings with a specific 3D deprojection stage, which we will present next (§3.1), to embed them in the continuous 3D query space and allow 3D convolutions and reasoning. Figure 2 provides an overview of the approach.

3.1. Depth Feature Encoder

Recent literature has shown the robustness of hierarchical feature encoding for both 2D object detection [20] and 3D reconstruction [43] tasks. We adopt a U-Net [37]-like feature encoder which contains four downscale and four upscale convolutional operations, and use two feature levels, one coarse and one fine, for the depth feature encoding. Given a pair of depth images $\mathcal{D}_i \in \mathbb{R}^{res \times res}$ where $i \in \{s\text{: source}, t\text{: target}\}$, the embedded features of the second and fourth upscale operations will be fed respectively to the subsequent coarse and fine inference path. We note these feature maps $\mathcal{F}^j, j \in \{c\text{: coarse}, f\text{: fine}\}$, where $\mathcal{F}^j \in \mathbb{R}^{df^j \times res^j \times res^j}$, with *df* the dimensionality of pixel-aligned features.

Feature deprojection. Given a depth image $\mathcal{D}(x, y)$, the integer location $vox(x, y, z) \in \mathbb{Z}^3$ of occupied points in the grid can be computed using a deprojection operation [45]:

$$\mathcal{F}_{3D}(x, y, z) = \mathcal{F}(x, y), \text{if } \mathcal{D}(x, y) \neq \emptyset, \tag{1}$$

$$\text{where } vox(x, y, z) = (x_{\mathcal{D}}, y_{\mathcal{D}}, \lfloor \frac{\mathcal{D}(x, y) - z_{\text{near}}}{dl_{\text{map}}} \rfloor), \tag{2}$$

where z_{near} is the lower bound of depth in view volume of camera, dl_{map} is the cell size of grid, $x_{\mathcal{D}}/y_{\mathcal{D}}$ is x/y-coordinate in depth image, $\mathcal{D}(x, y)$ denotes the depth value at pixel (x, y) and $\lfloor\ \rfloor$ stands for floor operation. The deprojected 3D feature $\mathcal{F}_{3D}$ allows us to compute a cost between source and target frames for the following 3D scene flow estimation. For front-viewed points $\mathcal{V}(x, y, z)$, we append projective signed distance $PSDF(x, y, z) = \mathcal{D}(x, y) - z_{near} - (vox(z) + 0.5)dl_{map}$ along depth-viewing, into deprojected feature $\mathcal{F}_{3D}(x, y, z)$ as [54].

3.2. Coarse-Dense Flow Module

This module is designed to prepare a dense 3D representation which can be used to compute classic cost volume [42] for flow estimation. Given coarse level feature $\mathcal{F}^c_{3D}$, we pad 0 for un-viewed grid points,

$$\mathcal{F}^c_{3D}(x, y, z) = 0, \text{if } \mathcal{D}(x, y) = \emptyset \tag{3}$$

The raw 3D feature $\mathcal{F}^c_{3D}$ is still not able to handle ambiguity along depth direction, because only the frontmost point

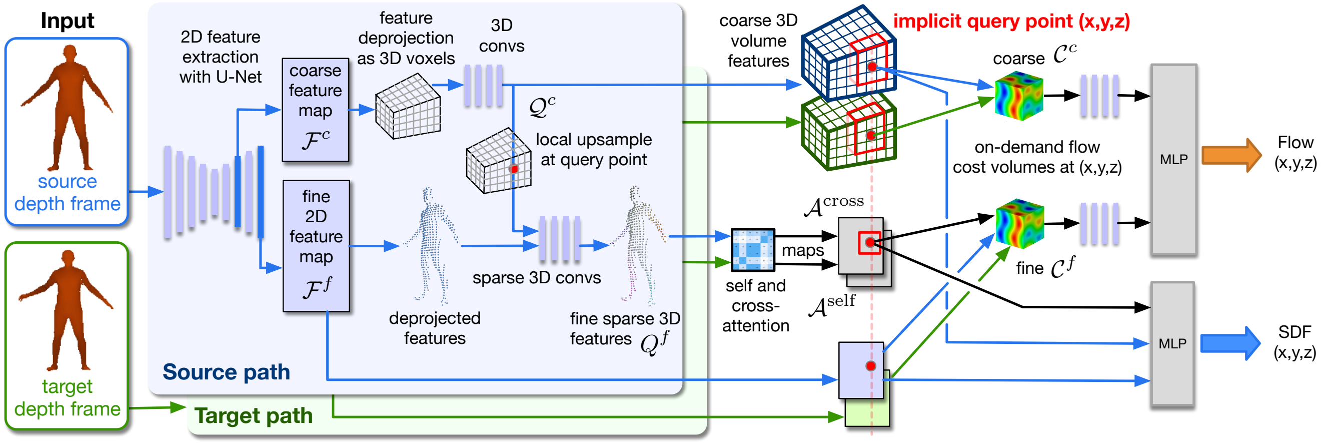
Input
source depth frame
target depth frame
2D feature extraction with U-Net
coarse feature map $\mathcal{F}^c$
fine 2D feature map $\mathcal{F}^f$
feature deprojection as 3D voxels
3D convs
$\mathcal{Q}^c$
local upsample at query point
deprojected features
sparse 3D convs
fine sparse 3D features $\mathcal{Q}^f$
Source path
Target path
coarse 3D volume features
implicit query point (x,y,z)
coarse $\mathcal{C}^c$
on-demand flow cost volumes at (x,y,z)
$\mathcal{A}^{\text{cross}}$
maps
self and cross-attention
$\mathcal{A}^{\text{self}}$
fine $\mathcal{C}^f$
MLP
Flow (x,y,z)
MLP
SDF (x,y,z)

Figure 2. Overview of the approach and network architecture.

along depth is filled with a meaningful feature. We therefore follow the idea of [7] to propagate features in the 3D grid with dense 3D-convolutional layers.

$$\mathcal{Q}^c = \texttt{Dconv3D}(\mathcal{F}^c_{3D}) \tag{4}$$

Note that [7] voxelizes a point cloud input into 3D grid with 1/0 values, and applies 11 costly 3D-convolutions from such 1/0 values. We propose to first encode features with 2D-convolutions, then deproject them into the 3D grid, where 6 3D-convolutional layers are applied to the deprojected features, which yields a more efficient scheme.

After our dense operations, $\mathcal{Q}^c_i$ is a dense 3D scene representation for source and target frames. Given an arbitrary query point $p(x, y, z)$ in the source scene, the difference cost from source to target scene can be computed as follows, and illustrated in Figure 3:

$$\mathcal{C}^c_{s\to t}(x, y, z, h, v, d, \delta^c) = \\ \texttt{tri}\{\mathcal{Q}^c_t(x + h\delta^c, y + v\delta^c, z + d\delta^c)\} - \texttt{tri}\{\mathcal{Q}^c_s(x, y, z)\} \tag{5}$$

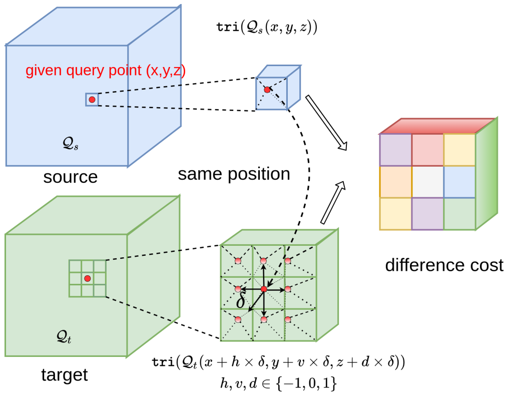
$\texttt{tri}(\mathcal{Q}_s(x, y, z))$
given query point (x,y,z)
$\mathcal{Q}_s$
source
same position
difference cost
$\mathcal{Q}_t$
δ
target
$\texttt{tri}(\mathcal{Q}_t(x + h \times \delta, y + v \times \delta, z + d \times \delta))$
$h, v, d \in \{-1, 0, 1\}$

Figure 3. Implicit cost computation.

where δ^c is a small shift length depending on resolution, $h, v, d \in \{-1, 0, 1\}$ are shift values along x, y, z-axis and **tri** is the tri-linear interpolation operation. Unlike PWC-Net [42] which computes a correlation cost between source and target scene, we opt for a difference cost, an immutable operation, because we wish to compute both forward and backward flow to enforce cycle consistency.

Given the cost $\mathcal{C}^c_{s\to t} \in \mathbb{R}^{f\times3\times3\times3}$, we compute the point-wise flow embedding with $3 \times 3 \times 3$ convolutional layers `Conv3x3x3`,

$$\mathcal{Q}^c_{s\to t}(x, y, z) = \texttt{Conv3x3x3}(\mathcal{C}^c_{s\to t}) \tag{6}$$

Note that `Conv3x3x3` is applied to the point-wise cost value, unlike previous 3D-convolutions spanning the whole grid with padding and striding, so `Conv3x3x3` is simple and cheap to compute. The embedded feature will be fed into the flow decoder to predict 3D scene flow, along with cost feature computed with fine features processed by the following fine-sparse and attention modules.

3.3. Fine-Sparse Module

Since dense 3D convolution is very time- and memory-consuming, especially for high-resolution grids, we follow the idea of completion work [19] and use Sparse Convolutions [10, 11]. We use the occupied location vox^f and the feature values $\mathcal{F}^f_{3D}$ as inputs to sparse operations without any padding. Sparse convolutions affect any observed point in the receptive field of the applied kernel, with the inherent limitation that it accounts only for occupied points. For the completion task however, we do need to propagate information to unoccupied regions within the receptive field. To handle this, [19] applies dense 3D convolutions after sparse convolution encoding and outputs a signed distance field at full resolution, which would adversely increase the memory cost in our case. Instead, we propose a classic coarse-to-fine strategy in order to extract high-frequency sparse features Q^f, which achieves this task with low resources, as follows:

$$\mathcal{Q}^{c-up} = \texttt{tri}\{\mathcal{Q}^c(vox^f(x,y,z))\} \tag{7}$$

$$Q^f = \texttt{SConv3D}((\mathcal{Q}^{c-up} \oplus \mathcal{F}^f_{3D}), vox^f) \tag{8}$$

where $\mathcal{Q}^{c-up}$ is the up-sampled feature from the coarse-dense module, which contains the information from the unoccupied field and serves as an initialization for fine level features. $\oplus$ is the concatenation operation and SConv3D denotes the sparse 3D convolution operator. Since SConv3D is active only for observed points, $Q^f \in \mathbb{R}^{n \times dq}$ is indeed a point-wise feature where n is observed point number and dq is the feature dimension.

3.4. Self/Cross-Attention Module

Classic convolution operations, dense or sparse, only affect a specific receptive field of the size of the kernel. However, such a receptive field cannot cover the whole visible space, in other words, we cannot set the kernel size as res due to the memory cost. Layer by layer convolutional operations may alleviate this phenomenon, but a more straightforward approach to ensure that longer range similarities or symmetries are exploited is to use an all-to-all attention mechanism. Moreover, corresponding features for a moving point from source to target could be identified and analyzed by this technique. We thus leverage both self- and cross-attention in our architecture. Recall that, in a generic setup, attention usually concerns querying a generic set of features Q_{qu}, given a key/value set Q_{kv}, attention can be described as follows:

$$\begin{aligned} m_{qu} = Q_{qu}W_{qu}, m_{va} = Q_{kv}W_{va}, m_{ke} = Q_{kv}W_{ke}, \\ q_{att} = \texttt{Softmax}(m_{qu}m^T_{ke}/\sqrt{da}), \\ A(Q_{qu}, Q_{kv}) = \texttt{Lin}(q_{att}m_{va}) \end{aligned} \tag{9}$$

where $W_{qu}, W_{va}, W_{ke} \in \mathbb{R}^{dq \times da}$ are three learnable parameters, da is the embedding dimension and Lin is one linear layer. In our case, in the direction from source to target, we set $Q_{qu} = Q_{kv} = Q^f_s$ when computing self-attention A^{self}_s, while we set $Q_{qu} = Q^f_s, Q_{kv} = Q^f_t$ when computing cross-attention $A^{\text{cross}}_{s\to t}$. Unlike point-to-patch [22] or patch-to-patch [50] attention mechanisms, our method considers all observed points' impacts on a query point.

We believe that symmetry and correspondence information of self- and cross-attention contribute not only to completion but also to the flow task. As in Sec. 3.2, we should define the cost value given attentions $A^{\text{self}}_s, A^{\text{cross}}_{s\to t}, A^{\text{self}}_t$ and $A^{\text{cross}}_{t\to s}$. But attention features A have the same sparsity structure as Q^f, *i.e.* $Q^f(vox(x,y,z))$ and $A(vox(x,y,z))$ exist only if $\mathcal{D}(x,y) > 0$. Such a cost should also be defined with unoccupied points on the depth because the trajectory of a moving point can cover several unoccupied points.

Sparse Feature Densification: We thus simply project sparse 3D features A back to 2D pixel-wise feature $\mathcal{A}$ and pad 0 for un-observed pixels for the reason of keeping continuity along depth direction.

We collect high-resolution feature maps $\mathcal{Q}^h = \mathcal{F}^f \oplus \mathcal{A}^{\text{self}} \oplus \mathcal{A}^{\text{cross}}$, and follow the process as in Sec. 3.2 to compute high-frequency cost $\mathcal{C}^f$ and flow features,

$$\begin{aligned} \mathcal{C}^f_{s\to t}(x,y,h,v,\delta^f) = \\ \texttt{bi}\{\mathcal{Q}^h_t(x+h\delta^f, y+v\delta^f)\} - \texttt{bi}\{\mathcal{Q}^h_s(x,y)\} \end{aligned} \tag{10}$$

$$\mathcal{Q}^f_{s\to t}(x,y,z) = \texttt{Conv3x3}(\mathcal{C}^f_{s\to t}) \oplus z \tag{11}$$

where Conv3x3 is 3×3 convolutional layers to embed 2D projected flow. To solve the depth ambiguity, the high-frequency flow feature $\mathcal{Q}^f_{s\to t}$ is appended with the depth value z of the query point in source.

3.5. Shape/Flow Decoder

Given all aforementioned extracted features and query point position $p(x,y,z)$, we prepare two multilayer perceptrons [12] Ψ, Φ to predict SDF and flow simultaneously.

$$\text{sdf}(x,y,z) = \Psi(\mathcal{F}^f, \mathcal{Q}^c, \mathcal{A}^{self}, \mathcal{A}^{cross}, z) \tag{12}$$

$$\text{flow}_{s\to t}(x,y,z) = \Phi(\mathcal{Q}^c_{s\to t}, \mathcal{Q}^f_{s\to t}) \tag{13}$$

where sdf $\in \mathbb{R}$, flow $\in \mathbb{R}^3$.

3.6. Training Loss

Φ computes both forward flow flow$_{s\to t}$ and backward flow flow$_{t\to s}$ to enforce cycle consistency. In parallel, Ψ predicts SDFs in both source and target frame. Finally, we train the whole architecture with squared loss between the prediction and ground truth sdf^{gt}, flow^{gt}.

$$\begin{aligned} \mathcal{L} = \sum_i \left(\sum_{bs=1}^{n^i} \alpha \|\text{sdf}_{bs} - \text{sdf}^{gt}_{bs}\|^2 \right) \\ + \sum_{di} \left(\sum_{bf=1}^{n^{di}} \beta \|\text{flow}_{bf} - \text{flow}^{gt}_{bf}\|^2 \right) \end{aligned} \tag{14}$$

where $i \in \{s,t\}$, $di \in \{s \to t, t \to s\}$ and n^i, n^{di} are query points numbers for SDF and flow estimation. α, β are coefficients to balance the loss $\mathcal{L}$.

4. Experiments

We provide quantitative and qualitative results on DFAUST [6] and CAPE [26] (Section 4.1). Additional results on THUman3.0 [41] are shown in the supplemental material. We first present comparisons with other methods in Section 4.3. We further demonstrate the contributions of our approach core components with ablation studies in Section 4.4. Supplemental material is available at https://hal.inria.fr/hal-04045719.

4.1. Datasets and Metrics

We consider real world scan data from DFAUST [6] for undressed humans and from CAPE [26] for dressed humans where raw data is fitted with SMPL [24]. Both datasets are used for training. Real scans often contain holes and noisy observations, which make them difficult to use as ground truth. We use instead the SMPL-fitted data to train our networks as in [31, 58]. We posit that such data provided by DFAUST and CAPE contains sufficient surface details for our network to capture at the local level, and regress and generalize from the local high frequency patterns, in the input depth images, independently of each other. Given 3D shape geometries, depth images are rendered in resolution 256^2 with a fixed viewpoint for dynamic sequences by using PyTorch3D [35]. Signed distances are pre-computed from the watertight meshes. During test time, depth images are rendered from the raw scan data to preserve measurement noise and to evaluate robustness.

For the shape completion, we evaluate methods shape by shape with the Intersection Over Union(IoU) and the Chamfer distance metrics, and we follow [7, 58] for that purpose. For the flow estimation, we use the end-point-error (EPE), similarly to [22,31], and with respect to 2 different schemes: Tracking errors evaluate the average EPE between the first frame in a sequence and all the other frames, whereas the pairwise flow errors measure the average EPE between successive frames of around 70ms in the sequence.

4.2. Training

We use for training 2 male and 2 female characters from CAPE, where each character is dressed with 2 or 3 different clothing styles. We also include 2 male and 2 female characters from DFAUST, for a total of 8 characters, 258 short-term and 252 long-term pairs of depth images. The short-term interval is around 70ms while the longer one 200ms.

During training, pre-computed signed distances for query points are given as supervision. As mentioned in [38, 39], the sampling strategy has a major impact on the reconstruction quality. We follow a threefold strategy that has proven efficient in practice. For a given scene, we sample 1600 points near the surface to capture the fast evolution of the SDF in such regions. We additionally sample 400 points in a bounding box to avoid the ghost artifacts that can appear when focusing only on near-surface regions. We note that such points satisfy the condition that the distance is farther than 2cm from the surface, and we consider these points fixed during flow estimation. Moreover we sample 400 surface points for both shape and flow estimations. Before computing the loss at each iteration, we rescale the SDF by factor 50 and flow by factor 10 for numeric stability. In addition the rescaled SDF is truncated to value 1. We also set coefficient $\alpha = 1$ everywhere, $\beta = 1$ for surface points and $\beta = 0.1$ for 400 points whose flows are 0.

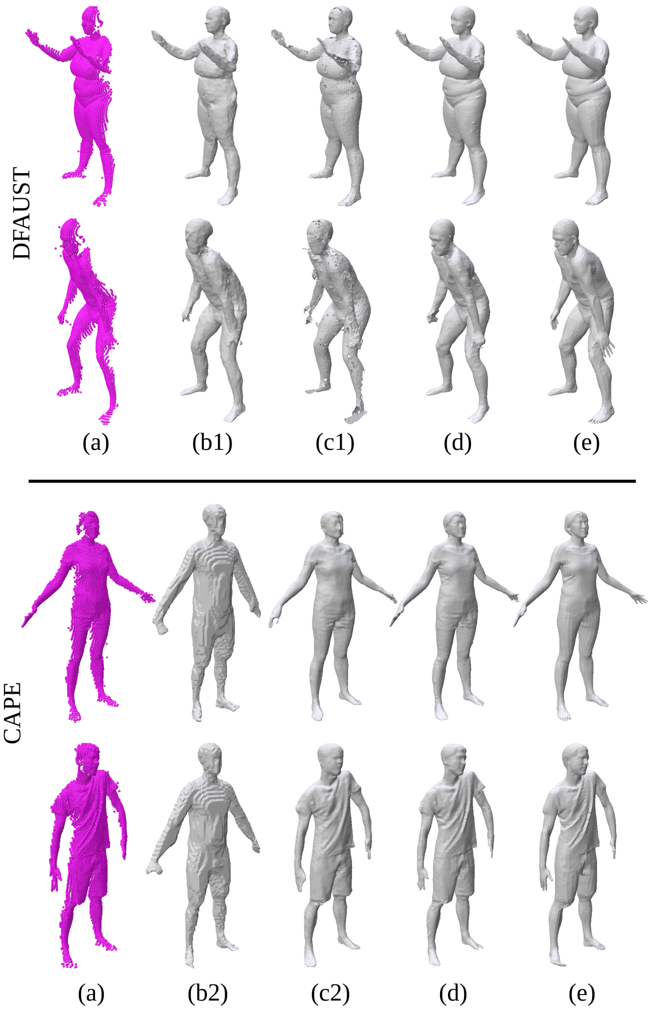

Figure 4. Shape completions. From left to right, we show (a) partial inputs, completions of (b1) IF-Net [7], (b2) ShapeFormer [51], (c1) SeedFormer [59], (c2) STIF [58], (d) ours and (e) ground truth.

4.3. Comparisons

DFAUST: We compare our method with baseline methods that use different scene representations: 1. *Mesh*, 3D-CODED [13] deforms a mesh template. 2. *Implicit* ONet [28], IF-Net [7], OFlow [30], STIF [58] complete shapes using implicit representations. 3. *Voxels* 4DComplete [19] estimates shape and flow in a regular voxel grid. 4. *Points* SeedFormer [59] completes directly partial point clouds. All the compared methods are trained with our training data, except OFlow which operates under its provided pre-trained model. Results from SeedFormer were meshed with the Ball Pivoting method [4], similarly to [18]. However, the resulting meshes are not watertight, see Fig. 4, which invalidates the IoU metric for SeedFormer. Results of Onet, IF-Net, STIF, OFlow and our method were meshed using the Marching Cubes [16, 25], as in [28]. Since Oflow [30] processes sub-sequences of 17 frames by construction, we report the tracking EPE for such sub-sequences in Tab. 2. In total, we evaluate on 31 sub-sequences of 2 unseen characters and 2 seen characters performing unseen motions in DFAUST. We observe that, in general, implicit representations perform better than other representations. We also note that our method consistently outperforms all other methods w.r.t. all metrics. In Fig. 4, it can be observed that our method preserves more surface detail than other methods.

Method	IoU ↑	Chamfer-L1 ↓	EPE ↓	Represent.
3D-CODED [13]	0.339	5.412	0.572	Mesh
ONet [28]	0.349	5.365	-	Implicit
IF-Net [7]	0.826	1.197	-	Implicit
OFlow [30]	0.708	2.510	0.374	Implicit
4DComplete [19]	0.679	3.047	1.373	Voxels
STIF [58]	0.850	1.133	-	Implicit
SeedFormer [59]	-	1.056	-	Points
Ours	**0.862**	**1.029**	**0.212**	Implicit

Table 2. DFAUST comparisons with chamfer $\times 10^{-2}$ and EPE $\times 10^{-1}$. The flow is evaluated with the tracking EPE errors (see Sec. 4.1) over 17-frame subsequences as in OFlow [30].

CAPE: Tab. 3 aggregates quantitative results provided in NPMs [31], in addition to the results we obtained with STIF, SeedFormer and ShapeFormer [51]. ShapeFormer was also trained with our data. In NPMs the evaluation is conducted on sub-sequences of 17 frames from 4 sequences of 4 identities. We report both tracking EPE and pair-wise flow EPE when relevant. Our method again achieves best performance for shape completion. It also appears very competitive for tracking performance, and outperforms NPMs on flow estimation, as illustrated in Fig. 1 and 5.

Method	IoU ↑	Chamfer-L2 ↓	EPE ↓	Represent.
OpenPose+SMPL	0.68	0.243	2.82/	Model
OFlow [30]	0.55	0.755	2.65/	Implicit
IP-Net [5]	0.82	0.034	2.52/	Model
NPMs [31]	0.83	0.022	**0.74**/0.43	Model
ShapeFormer [51]	0.48	0.824	-	Explicit
STIF [58]	0.85	0.035	-	Implicit
SeedFormer [59]	-	0.037	-	Points
Ours	**0.87**	**0.017**	0.75/**0.38**	Implicit

Table 3. CAPE comparisons. We follow the setup of NPMs to bound results in a normalized space and to evaluate shape reconstruction and motion tracking over sub-sequences of 17 frames. Our metrics are IoU, Chamfer-L2 $\times 10^{-3}$ and tracking/pair-wise flow EPE $\times 10^{-2}$ (see Sec. 4.1).

4.4. Ablation

The ablation tests were conducted on 76 pairs of depth images from 6 unseen characters and 2 seen characters performing unseen motions. Tab. 4 evaluates 4 configurations:
(i) Baseline: the depth encoder and shape decoder only when single-frame is given.
(ii) 3D flow: the coarse-dense module and the cost volume for the flow estimation are added.
(iii) Fine-sparse: Sparse convolutions for high-frequency features are hierarchically added.
(iv) Self/cross attention: Self/cross attention are added to enforce both spatial and temporal consistency.

The different components all contribute to improve the results. We note that the flow improves IoU the most, while the main impact on Chamfer distance comes from the fine-sparse strategy. These results should nevertheless be considered with the fact the ground truth shapes are fitted SMPL models that do not exactly fit original surface details. In that respect IoU is a more robust volumetric metric than the surface based Chamfer distance.

Method	IoU ↑	Chamfer-L1 ↓	EPE ↓
baseline	0.839	1.146	-
+3D dense flow	0.848	1.121	0.137
+fine-sparse	0.854	1.051	0.133
+attention	**0.861**	**1.047**	**0.121**

Table 4. Ablation tests with Chamfer-L1 $\times 10^{-2}$ and EPE $\times 10^{-1}$.

5. Conclusion

We have presented a novel learning based approach that combines implicit human shape completion with implicit motion flow estimation given 2 depth images. Besides the combination of shape and flow features, the approach builds on 2 components with a coarse-dense to fine-sparse strategy, as well as an attention mechanism that helps exploiting the correlation between features inside and across frames. Ablation tests demonstrate the respective benefits of these components to the shape completion task. We also compare with other methods on 2 standard real human body datasets. They show that our approach consistently outperforms the other methods with more precise reconstructions, and advances therefore the state of the art in human shape completion with potentially more practical applications.
Limitations and Future Work. Although high-frequency details in front-view are better extracted by our method, it might make the unobserved part noisy for challenging poses, e.g. squat. We believe this is because some components, such as sparse convolution and attention, focus on local patterns. Due to the lack of human topology prior, strong occlusion is still challenging for our method. This points to several interesting future directions, such as combining implicit and model-based methods into a virtuous cy-

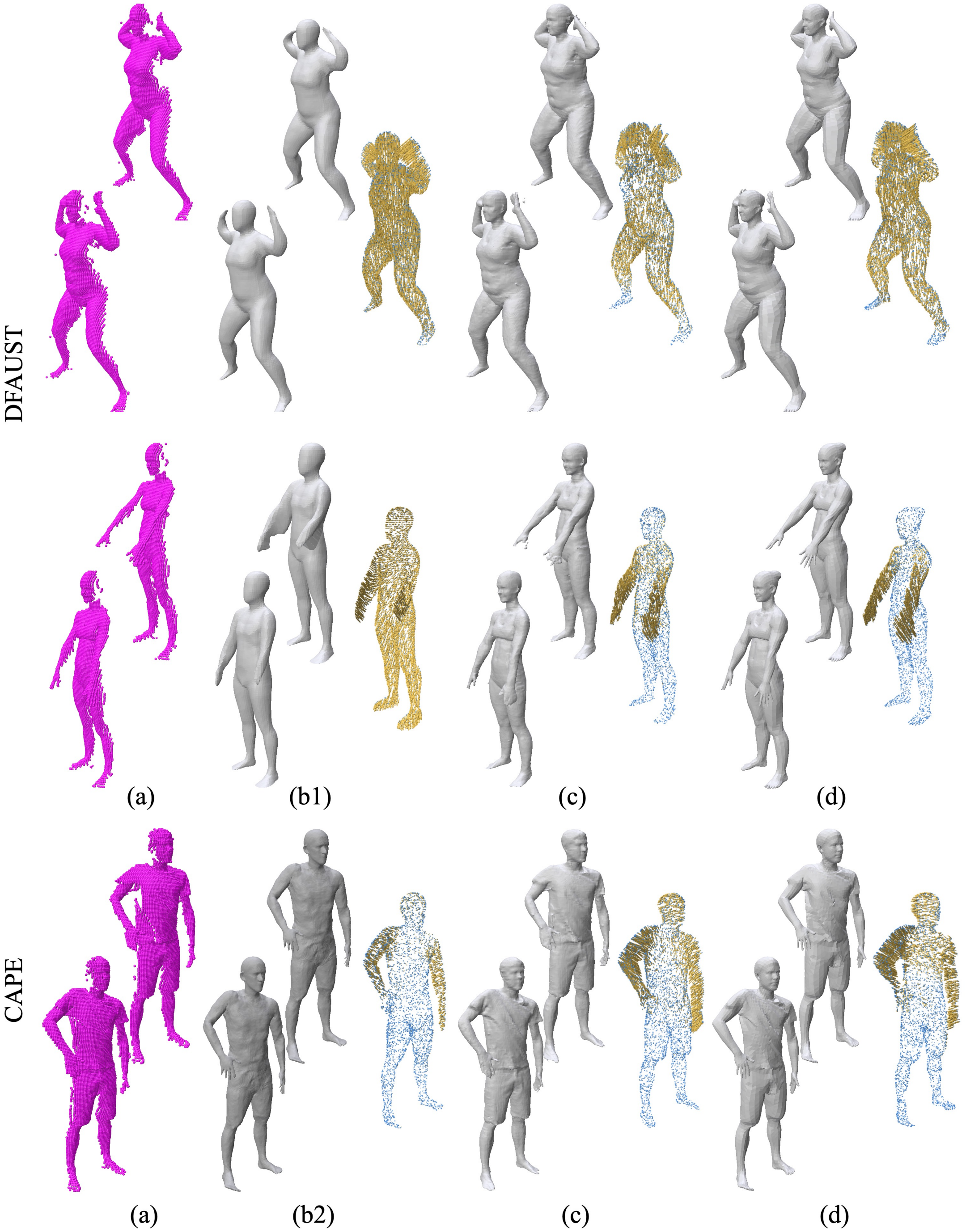

Figure 5. Shape completions and flow estimation. From left to right, we show (a) partial inputs, shape completions and flow estimations of (b1) OFlow [30], (b2) NPMs [31], (c) ours and (d) ground truth.

cle. Moreover we note the interest of considering temporal sequences with arbitrary lengths. Currently the flow extraction can provide critical low level information when dealing with more than 2 frames. In such a situation, the approach could be applied recursively, frame to frame, as an initial estimation to then drive a more global inference.

Acknowledgement. This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003).

References

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *Proceedings of the International Conference on 3D Vision*, 2018. 2

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics (TOG)*, 24(3):408–416, 2005. 2

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[4] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 6

[5] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *Proceedings of the European Conference on Computer Vision*. Springer, August 2020. 1, 2, 7

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 5, 6

[7] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 4, 6, 7

[8] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems*, 2020. 2

[9] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[10] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 4

[11] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 4

[12] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the International Conference on Machine Learning*, 2020. 2, 5

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision*, 2018. 6, 7

[14] David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Proceedings of the European Conference on Computer Vision*, 2012. 2

[15] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. H4d: Human 4d modeling by learning neural compositional representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3

[16] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003. 7

[17] Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[18] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6

[19] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the International Conference on Computer Vision*, 2021. 2, 4, 6, 7

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[21] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 3

[22] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5, 6

[23] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):220, 2014. 2

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transaction on Graphics (TOG)*, 34(6):248:1–248:16, Oct. 2015. 1, 2, 6

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987. 7

[26] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5, 6

[27] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. *arXiv preprint arXiv:1904.03278*, 2019. 2

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6, 7

[29] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[30] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 3, 6, 7, 8

[31] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the International Conference on Computer Vision*, 2021. 1, 2, 6, 7, 8

[32] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[33] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (TOG)*, 36(4):1–15, 2017. 2

[34] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. In *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, volume 34, pages 120:1–120:14, Aug. 2015. 1

[35] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 6

[36] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 2

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 3

[38] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 3, 6

[39] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3, 6

[40] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[41] Zhaoqi Su, Tao Yu, Yangang Wang, and Yebin Liu. DeepCloth: Neural garment representation for shape and style editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 5

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 4

[43] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[44] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[45] Hugues Thomas, Ben Agro, Mona Gridseth, Jian Zhang, and Timothy D Barfoot. Self-supervised learning of lidar segmentation for autonomous indoor navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14047–14053. IEEE, 2021. 3

[46] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[47] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021. 2

[48] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. In *Advances in Neural Information Processing Systems*, 2021. 2

[49] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Proceedings of the International Conference on Computer Vision*, 2011. 2

[50] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *Proceedings of the European Conference on Computer Vision*, 2020. 2, 5

[51] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3, 6, 7

[52] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017. 2

[53] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2

[54] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3

[55] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[56] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *Proceedings of the International Conference on 3D Vision*, 2018. 3

[57] Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu. Hybridfusion: real-time performance capture using a single depth sensor and sparse imus. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[58] Boyao Zhou, Jean-Sebastien Franco, Federica Bogo, and Edmond Boyer. Spatio-temporal human shape completion with implicit function networks. In *Proceedings of the International Conference on 3D Vision*, 2021. 2, 3, 6, 7

[59] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Proceedings of the European Conference on Computer Vision*, 2022. 3, 6, 7